MØDSCØ, a computer programm for the revised method of scoring

Vandaele, W.H.

Publication date:
1970

Document Version
Publisher's PDF, also known as Version of record

Link to publication in Tilburg University Research Portal

Citation for published version (APA):
Vandaele, W. H. (1970). *MØDSCØ, a computer programm for the revised method of scoring*. (EIT Research Memorandum). Stichting Economisch Instituut Tilburg.

Walter H. Vandaele

MØDSCØ, a computer programm for the revised method of scoring

[redacted]

Research Memorandum R 41

T software
V maximization
V Modsco

TILBURG INSTITUTE OF ECONOMICS
DEPARTMENT OF ECONOMETRICS

MØDSCØ, A computer program for the Revised Method of Scoring

by

Walter H. VANDAELE *

1. Purpose

To obtain maximum likelihood estimates of parameters in a non-linear model by maximizing the logarithm of the likelihood function of the parametervector $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$. The method of the iterative technique used is described in VANDAELE, Walter H. and S. R.CHOWDHURY, "A Revised Method of Scoring" [3].

In this research memorandum a comparison is made with the normal Fisher Method of Scoring †. So this computer program will also be discussed.

The programs are written in FØRTRAN II Language for an IBM 1620-II computer with F and K standard 8 and 9, and with a 1311 2 disk unit.

2. In order to allow the greatest possible flexibility in handling the input of the data and in specifying the ma-

---

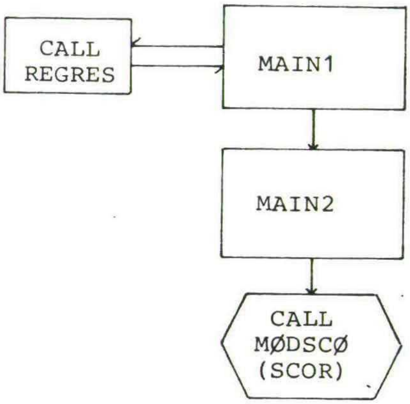
* The author likes to thank the University of Wisconsin Computing Center for providing him a non-linear least squares subroutine (GAUSHAUS). The GAUSHAUS program is based on the iterative technique of MARQUARDT, D.L. [1]. There can be drawn an analogue between the structure of these computer programs.

† For a description of the method and some references see VANDAELE, Walter H. [2, p. 34].

thematical model to be fitted on the data, MØDSCØ is written in the form of a subroutine. Normally the User must provide subroutines to generate the inverse of the Information matrix and to calculate value of the log likelihood function. Because in our research memorandum we have applied the Revised Method of Scoring on an autocorrelated model, with the usual assumption of a first order autoregressive scheme, the mentioned subroutines are made for that particular situation.

The following chart summarizes the function of the main program, the subroutine MØDSCØ and the User's subroutines which computes the inverse of the information matrix of the model.

MAIN - programs
(See Section 3)



Read observed function values Y, independent variables X, and starting parameter values TH (if to calculate: call REGRES). Certain constants NP, NT, MIT, EPS1, EPS2 ... are initialized. All these values are put in CØMMØN, and MØDSCØ (or SCØR) is called. When control is returned to MAIN2, another problem may be begun or the run terminated.

SUBROUTINE MØDSCØ (SCØR)
(See Section 4.5)

Performs maximum-likelihood estimates and provides printed output. At various times during the calculation, it calls the MØDT, MRES or INFØ subroutine. At the completion of the estimation procedure, control is returned to MAIN2.

SUBROUTINES

MØDT subroutine determines the calculated residuals (with the help of MRES) and the value of the log likelihood function for a choice of the parameters transmitted to it by MØDSCØ (SCØR). Subroutine INFØ calculates the inverse of the Information Matrix. Control is then returned to MØDSCØ (SCØR). The independent variable values, together with the needed information are obtained from CØMMØN.

CØMMØN

Contains X, Y, NP, NT, MIT, plus needed intermediate calculation results.

3 MAIN - programs

3.1 MAIN1-program

The functions of "MAIN1" program are

1. to supply the data and store them in CØMMØN;
2. to supply or calculate (via subroutine REGRES) the initial parameter value for the iterative procedure;
3. to initialize certain constants;
4. to call "MAIN2" where certain other constants will be initialized.

Read statements

a) READ 200, NP, IY, IRED, MIT, ISC

```
200 FØRMAT (10 I5)
```

NP — is an integer constant which is the number of parameters of the likelihood function to estimate without the σ (the standard-deviation of the residual term) and ρ (the autocorrelation parameter). Further in the program (see calling sequence of MØDSCØ), NK = NP + 2: the number of all the parameters, σ and ρ included. (See 4.1. The MØDSCØ calling sequence). NP must satisfy $0 < NP \leqslant 5$, $NP < NT$ (NT: the numbers of observations).

IY — is an integer constant which is the number of the sector on the disk where the Y-variable data can be found. †
If the data are supplied by cards, any number can be given to IY.

IRED — is an integer constant:
= 0 : the Y and X variables must be read from disk
= 1 : the Y and X variables are punched on cards.

If IRED = 0 or 1, then in the MAIN1 program, with the help of a subroutine REGRES, the initial value of the parameters are calculated

= 2 : the Y and X variables are on disk, and the initial parameter values are put on cards
= 3 : all information is put on cards: Y, X-values and initial value of the parameters

MIT — is an integer constant ($0 < MIT \leqslant 1000$) which is the maximum number of iterations to be performed. If the calculations have not been terminated for some other reason, they will be stopped when the number of iterations equals MIT.

---

† It is only necessary to indicate the sectornumber where the first 10 observations are on disk. In the program automatically the subroutine F2TCH is called 3 times, so that provision is made for timeseries up to 30 observations. With the parameters N, M, the proper selection of the observations is finally made.

ISC = 0 indicates that the Revised Method of Scoring will be used

= 1 the Fisher Method of Scoring will be applied.

b)
```
      READ   200, M, N
  200 FØRMAT (10 I5)
```

M, N — integer constants which indicate respectively the first and the last observation of the timeseries to be used.

c)
```
      READ   200, (IX(I), I = 1, NP)
  200 FØRMAT (10 I5)
```

IX — is an integer one-dimensional array containing the sector numbers on the disk of the explanatory variables. The same remark as that mentioned with IY, when IRED = 1 or 3, is valued.

d)
```
      READ   204, SH2, RØ
      READ   204,(BETA (I), I = 1, NP)
  204 FØRMAT (6 F 13.8)
```

BETA — is a real one-dimensional array containing the initial parameter values, except of σ^2 (SH2) and ρ (RØ). The σ, ρ and BETA's are then stored in that order in the real one-dimensional array TH before MØDSCØ (SCØR) is called. During the execution of MØDSCØ (SCØR), the most current parameter estimates are kept in TB, the previous values in TH.

e) If IRED = 1 or 3

```
     READ   201, (Y (J), J = M, N)
     DO 9   I = 1, NP
   9 READ   201, (X (J, I), J = M, N)
 201 FØRMAT (6 F 13.5)
```

Y — is a real one-dimensional array containing the vector of observed function values.

X — is a real matrix (NT x NP) containing the values of the explanatory variables.

3.2 Subroutine REGRES

REGRES is called from the main program when IRED is either 0 or 1. The FØRTRAN statement used is:

```
CALL REGRES   (NP, NT, Y, X, SH2, RØ, BETA, A, E, V, TE)
```

NP, NT, Y, X, SH2, RØ, BETA have the same meaning as that explained in MAIN1.

A — is a real one-dimensional array containing the vector of calculated Y-values.

E — is a real one-dimensional array of calculated disturbances: E = Y - A.

V — is a real one-dimensional array of normalized calculated disturbances.

TE — is a real one-dimensional array containing the principal diagonal elements of the $(X,X)^{-1}$ matrix. This vector is needed to calculate the t-values of the parameter estimates.

3.3 MAIN2-program

The MAIN1 program is connected with MAIN2 by a CALL LINK-statement. This program will initialize still some additional constants:

EPS1 is a real constant which is the value of the log likelihood function (without constant part †) $L_T(\theta)$ convergence criterion and is used to terminate the calculations based on the relative change of that value from one iteration to the next. More precisely, if at the completion of the i-th iteration, it is true that

$$\left| \frac{L_T(\theta^{(i)}) - L_T(\theta^{(i-1)})}{L_T(\theta^{(i-1)})} \right| \leq \text{EPS1} ,$$

then the calculations are terminated. In the program EPS1 = 10.0E - 05. If EPS1 is set equal to zero, this feature is disabled.

EPS2 is a real constant which is the parameter convergence criterion and is used to terminate the calculations based on the relative change in the parameter values from one iteration to the next one. If, at the completion of the i-th iteration, the following holds:

$$\left| \frac{\theta_j^{(i)} - \theta_j^{(i-1)}}{\theta_j^{(i-1)}} \right| \leq \text{EPS2}$$

for all j = 1 (1) NT, then the calculations are terminated. EPS2 is set equal to EPS1. This feature is disabled if EPS2 is set equal to zero.

---

† See [3, p. 12].

4. Subroutine MØDSCØ

4.1 MØDSCØ calling sequence

MØDSCØ is called from the MAIN2 program with a FØRTRAN statement of the form

CALL MØDSCØ (NK, EPS1, EPS2)

NK is an integer constant which is the total number of parameters of the likelihood function to estimate: σ (standarddeviation of the disturbance terms) and ρ (autocorrelation parameter) included

EPS1, EPS2 see 3.3 MAIN2-program.

In addition to the information transmitted through these arguments, the values of X, Y, TH, NP, NT, MIT, TE are assumed to be available from CØMMØN where they were put by the MAIN-programs.

4.2 Since the value of the likelihood function is needed at various times during the calculations, we provide subroutines to compute it. Whenever that value is needed for a particular value of the parameters TH, MØDSCØ simply transmits TH to that subroutine called MØDT.

SUBRØUTINE MØDT (NK, PAR, VAL)

NK as above

PAR is a one-dimensional array containing the parameter values for which the log likelihood is to be evaluated

VAL is a real variable indicating the value of the log likelihood function (without constant part).

A subroutine used in the MØDT subroutine is

SUBRØUTINE MRES (NK, PAR, A)

This program calculates the estimated y-values (A vector, a real one-dimensional array), as well as the estimated disturbances (E vector, a real one-dimensional array) for the values of PAR.

This subroutine is also separately used at the end of the MØDSCØ program, to calculate the final additional information (see 4.4).

4.3 Information matrix

During each iteration, the inverse of the Information matrix has to be evaluated. For that purpose a separate subroutine is used which is called in the MØDSCØ program with the calling statement.

CALL INFØ (NK, AA, B)

NK	as above
AA	is the 2 x 2 matrix of σ and ρ (See [3, p. 11])
B	is a NP x NP covariance matrix of regression parameters (BETA-parameters).

From CØMMØN subroutine INFØ assumes to have available the values of X, TH, NP, NT. Because of uniformity, a CØMMØN domain equal to that of MØDSCØ is used.

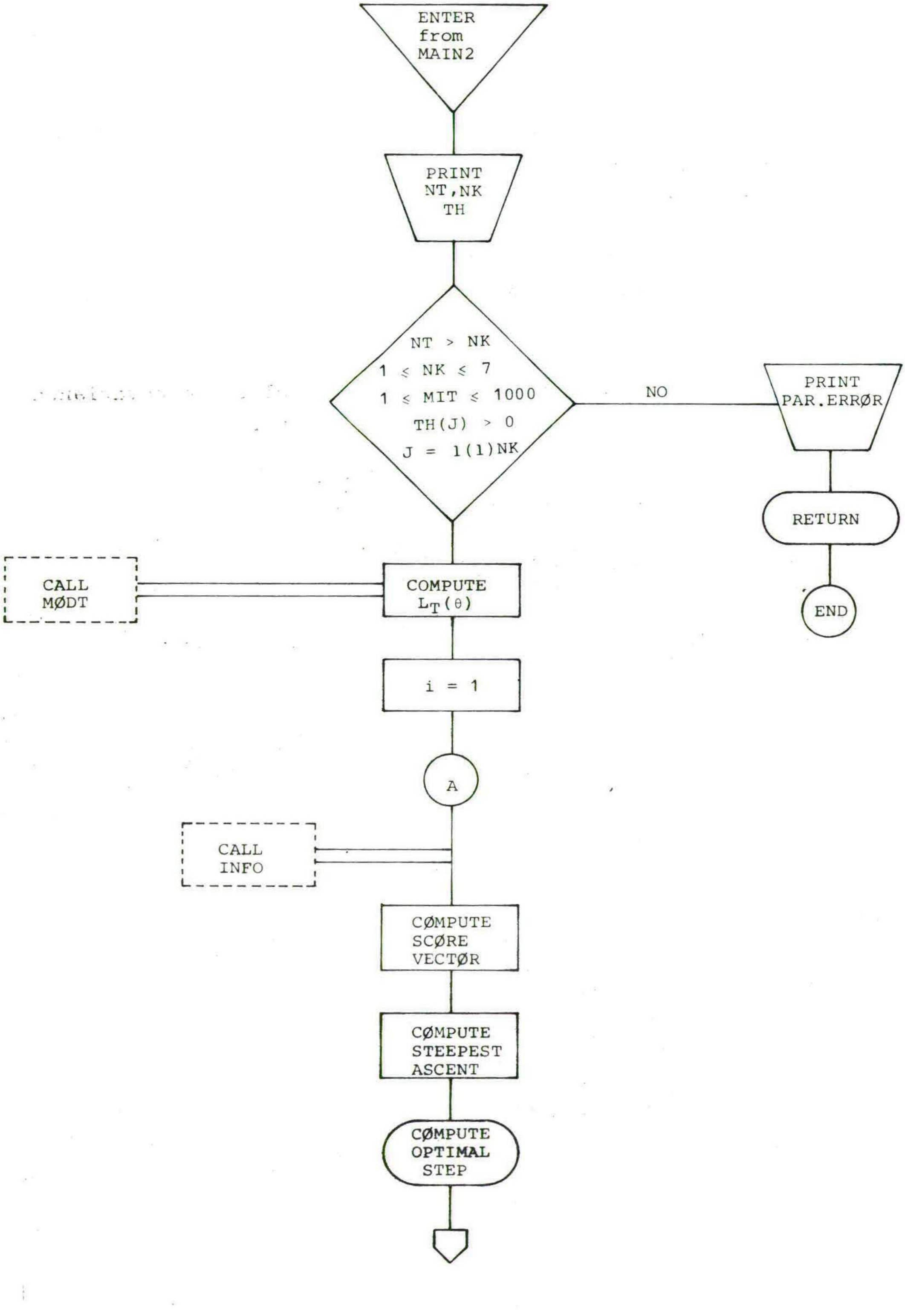
4.4 Final additional information

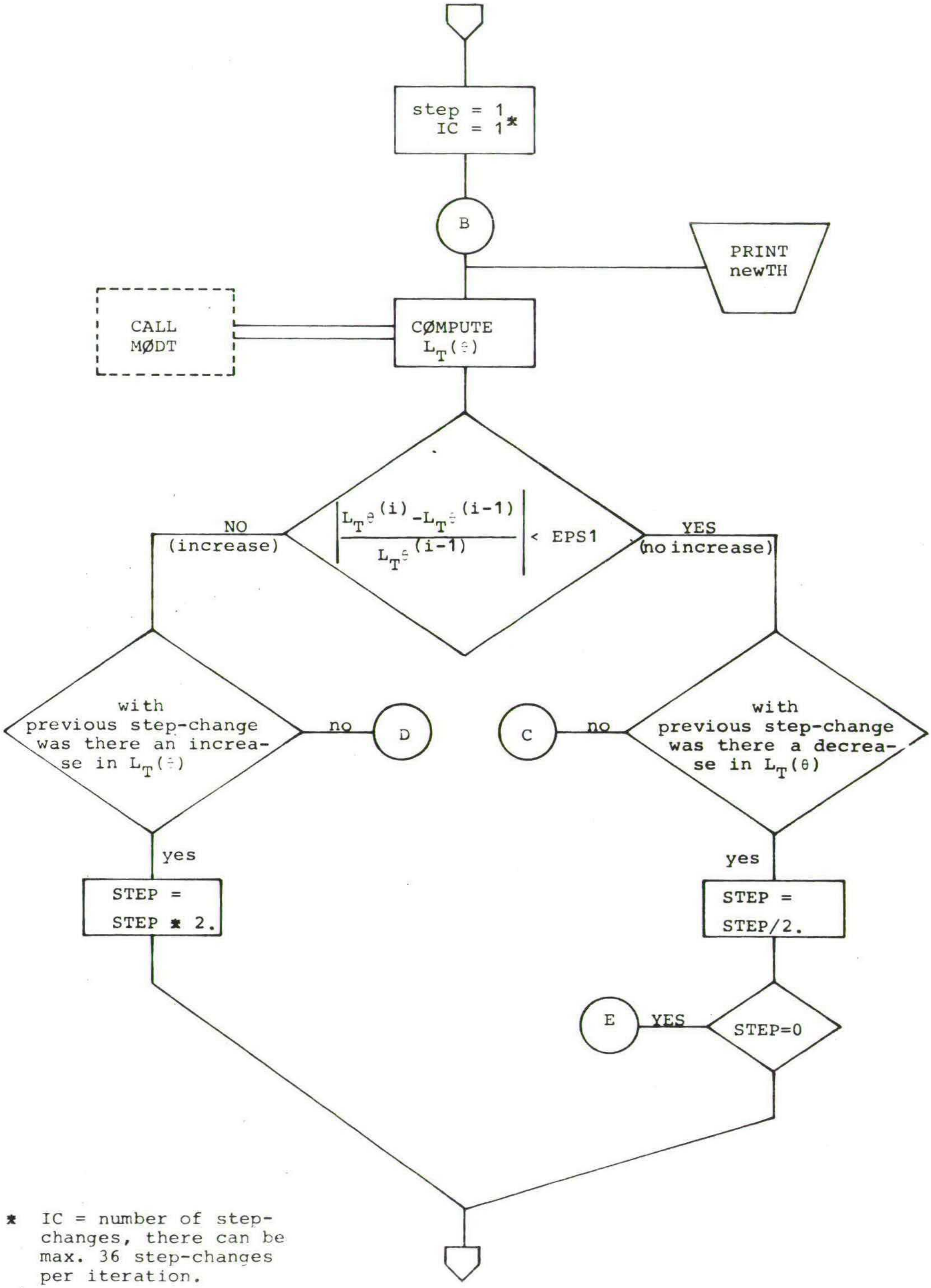
Before control is returned to MAIN2, the MØDSCØ subroutine supplies still some interesting additional information.

- Y-values: observed, and calculated ones, together with the residuals. The values are calculated in the MRES subroutine (See above);

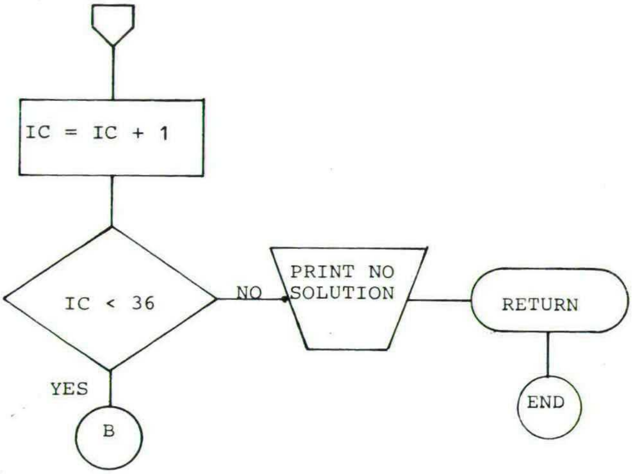
- The covariance matrix of σ and ρ, plus the covariance matrix of the regressioncoefficients.

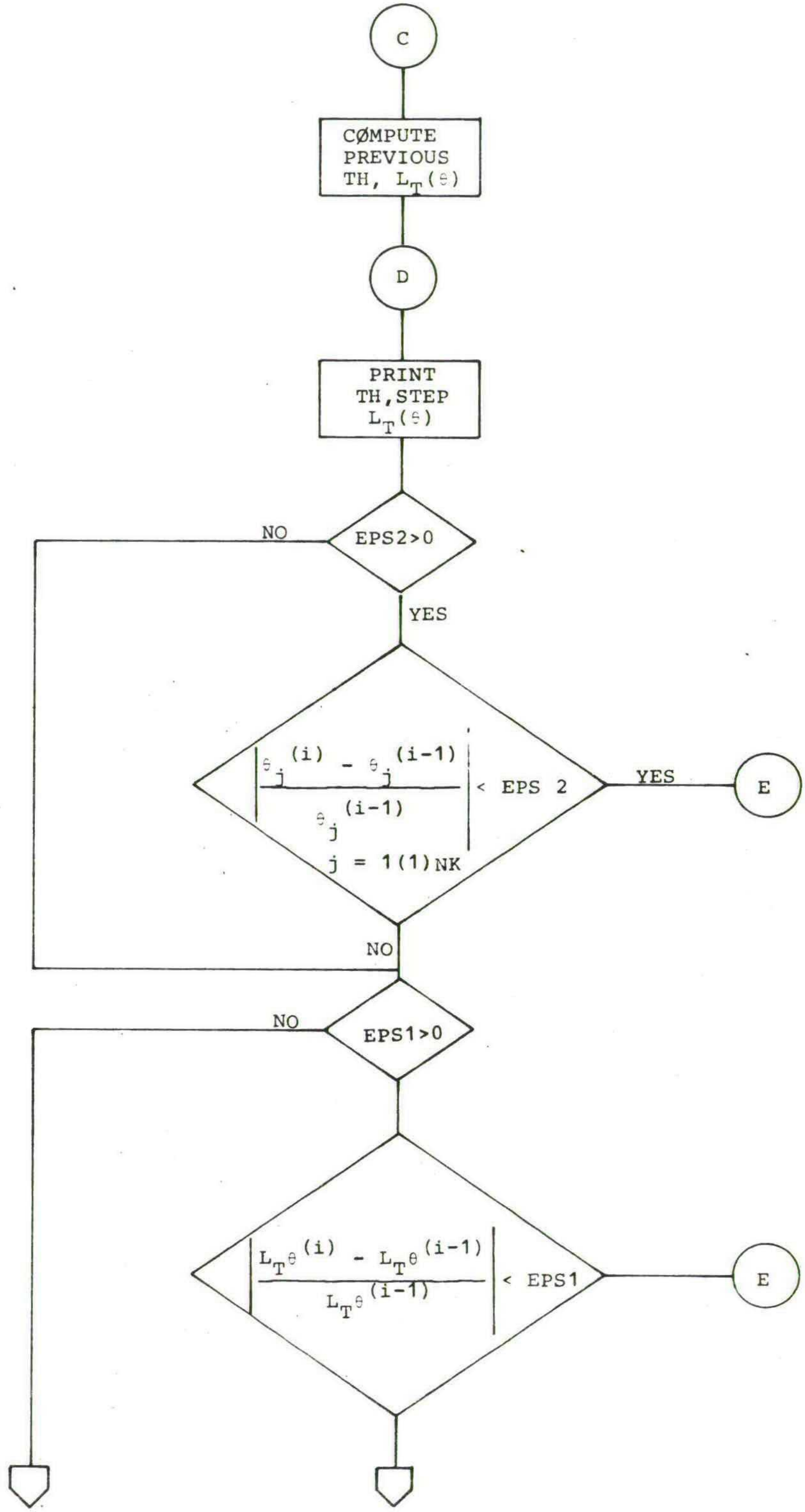
5 Flow Chart of MØDSCØ subroutine

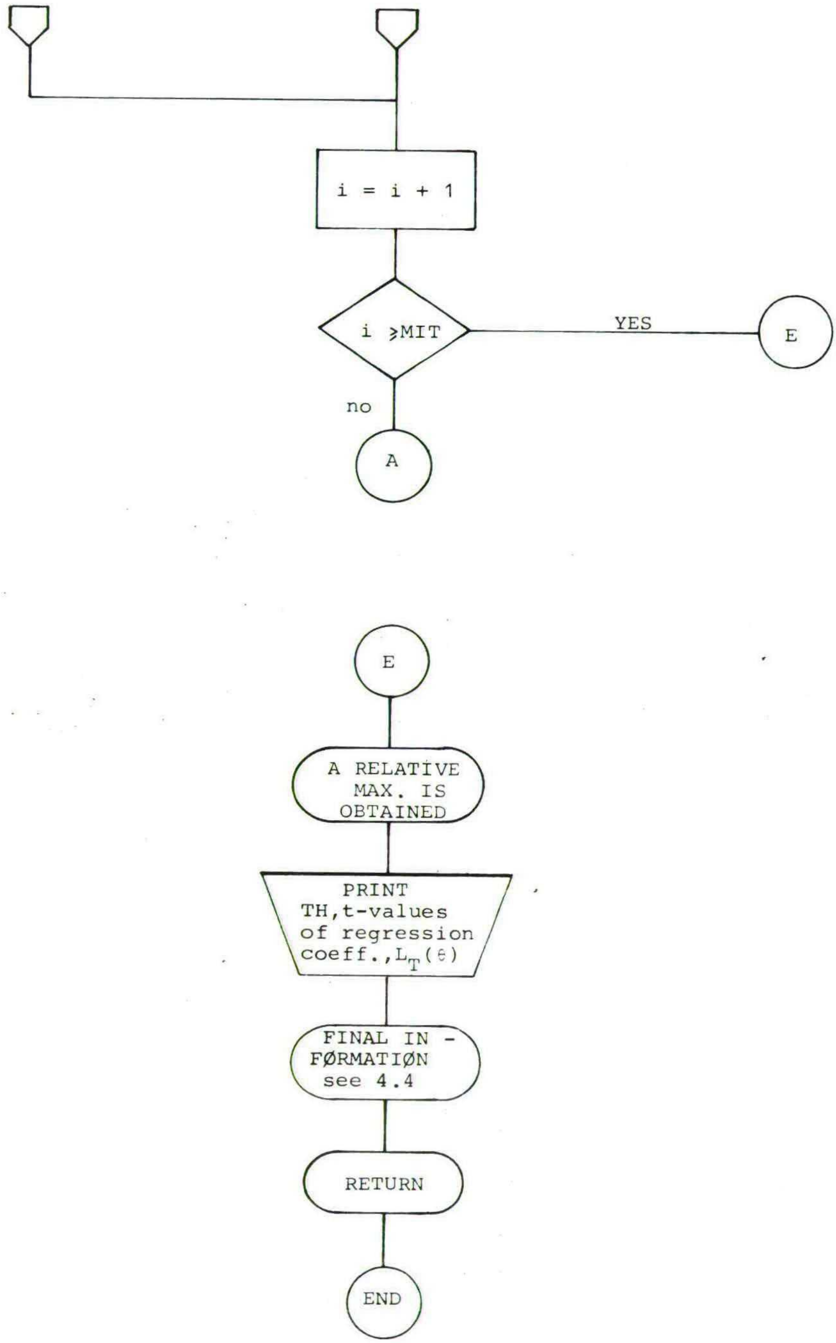




* IC = number of step-changes, there can be max. 36 step-changes per iteration.


IC = IC + 1
IC < 36
NO
YES
B
PRINT NO SOLUTION
RETURN
END


C
CØMPUTE
PREVIOUS
TH, $L_T(\theta)$
D
PRINT
TH,STEP
$L_T(\theta)$
NO
EPS2>0
YES
$\left|\frac{\theta_j^{(i)} - \theta_j^{(i-1)}}{\theta_j^{(i-1)}}\right| < EPS\ 2$
$j = 1(1)NK$
YES
E
NO
NO
EPS1>0
$\left|\frac{L_T\theta^{(i)} - L_T\theta^{(i-1)}}{L_T\theta^{(i-1)}}\right| < EPS1$
E


i = i + 1
i ≥MIT
YES
E
no
A
E
A RELATIVE MAX. IS OBTAINED
PRINT TH,t-values of regression coeff.,$L_T(\theta)$
FINAL IN - FØRMATIØN see 4.4
RETURN
END

6. SCØR program

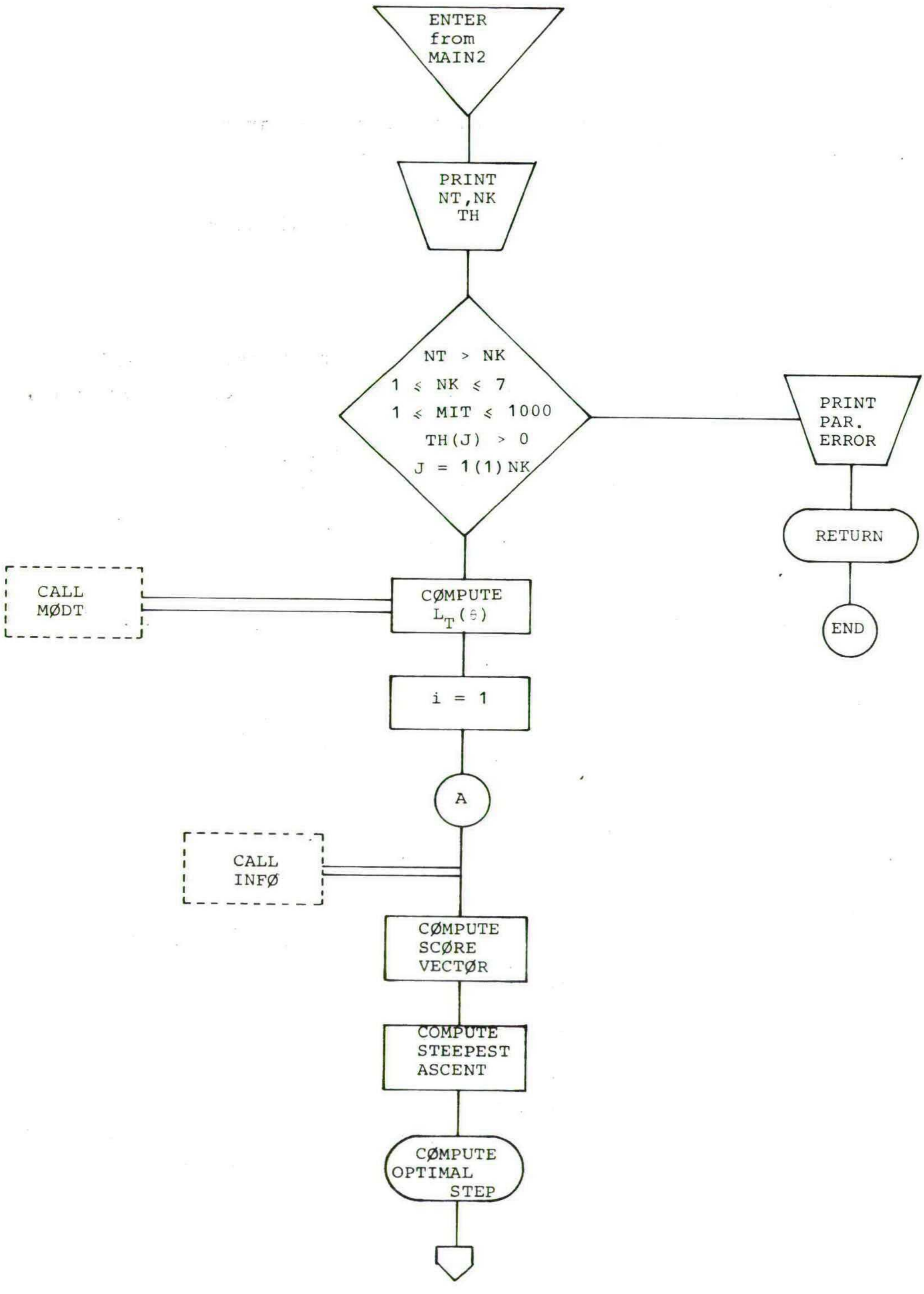
This subroutine calculates the parameter value with the Fisher's Method of Scoring.

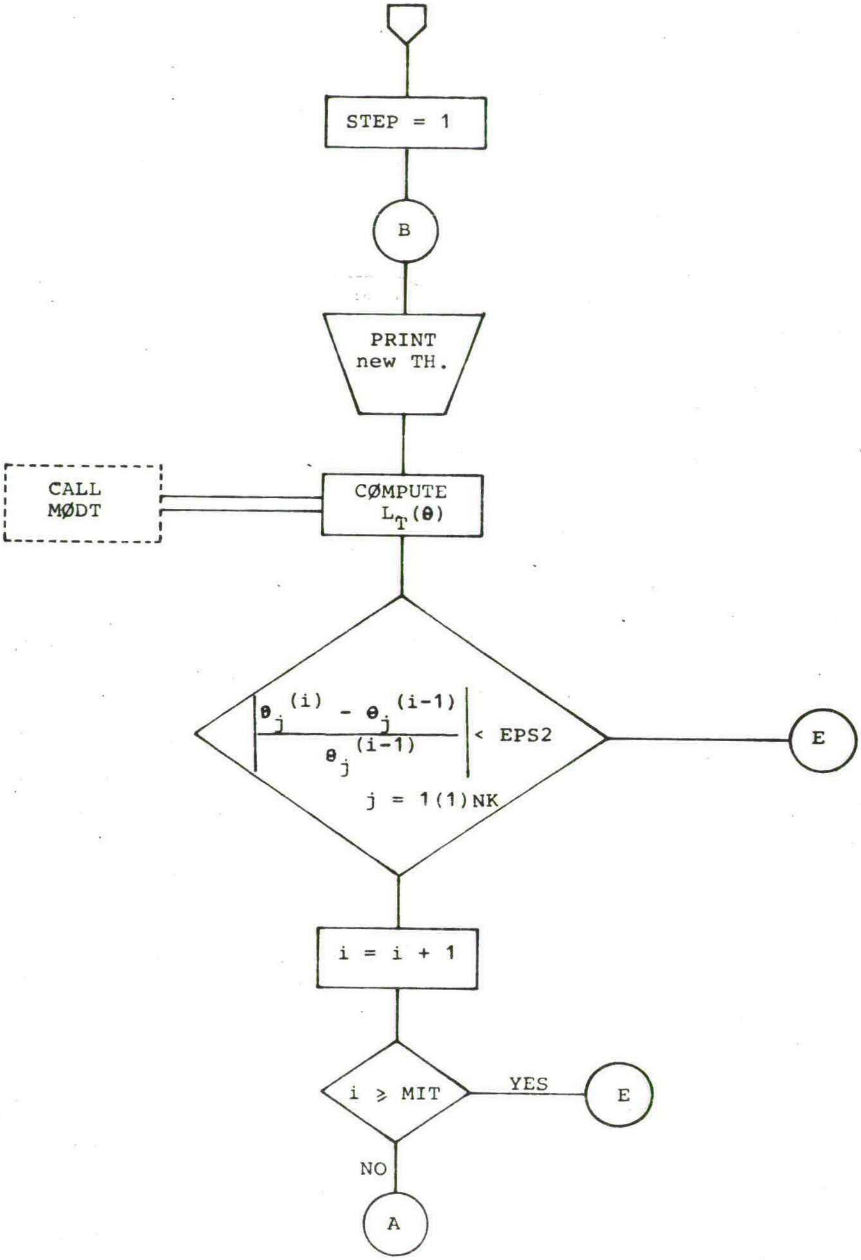
The calling sequence is the following:

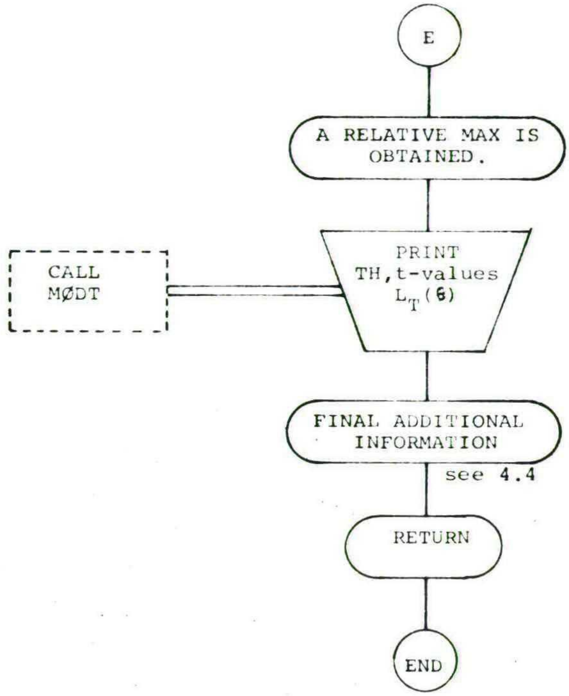
```
CALL    SCØR    (NK, EPS2).
```

For a description of these arguments, as well as the structure of this subroutine, we can refer to section 4.

7. Flow Chart of SCØR subroutine




STEP = 1
B
PRINT
new TH.
CALL
MØDT
CØMPUTE
$L_T(\theta)$
$\left| \frac{\theta_j^{(i)} - \theta_j^{(i-1)}}{\theta_j^{(i-1)}} \right| < EPS2$
j = 1(1)NK
E
i = i + 1
i ⩾ MIT
YES
E
NO
A


E
A RELATIVE MAX IS OBTAINED.
CALL MØDT
PRINT TH, t-values $L_T(\theta)$
FINAL ADDITIONAL INFORMATION
see 4.4
RETURN
END

8. Other subroutines used in the program

These programs are all available at the Computer Center of the Tilburg School of Economics, Social Sciences and Law:

F2TCH A subroutine which brings 10 floating-point numbers from a certain sector of the disk to the central memory.

MAPRI A vector and matrix printing subroutine.

INVE05 An inverse subroutine for a 5 x 5 matrix.

FLEM A subroutine used for multiplication of floating-point variables in double precision.

9. REFERENCES

[1] MARQUARDT, Donald W. "An algorithm for least-squares estimation of Nonlinear parameters", Journal Soc. Indust. Appl. Math., Vol. 11, June, 1963, nr. 2, pp. 431-441.

[2] VANDAELE, Walter H. Heteroscedastic Errors in Regression Analysis, Ter discussie nr. 6902, Katholieke Hogeschool, Economische Faculteit, May, 1969, 38 pp.

[3] VANDAELE, Walter H. and S. R.CHOWDHURY. A Revised Method of Scoring, Research Memorandum EIT 11, Tilburg Institute of Economics, 1970, 21 pp.

APPENDIX: Listing of the programs

```
C      MAIN1 PROGRAM - MODSCO-AUTOCORRELATED MODEL
C      FIRST PART
C
C      IRED = 0  X,Y FROM DISK
C      IRED = 1  X,Y FROM CARDS
C      IRED = 2  X,Y FROM DISK AND PARAMETERS FROM CARD
C      IRED = 3  X,Y  AND PARAMETERS FROM CARDS
C
C      ISC = 1 GO TO SCOR1
C      ISC = 0 GO TO MAIN2
C
       DIMENSION X(70,5),Y(70),E(70),TH(7),TE(5)
       DIMENSION IX(10),JX(10)
       DIMENSION Z(70),A(70),BETA(5),V(70)
C
       COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT,TE,ISC
C
  100 FORMAT (1H1,30H  ORIGINAL Y- AND X-VALUES(M=,I2,5H  ,N=,I2,1H))
  101 FORMAT (/)
  102 FORMAT (1H1)
  103 FORMAT (1H1,11H  Y-VALUES ,6X,8HOBSERVED,9X,10HCALCULATED,10X,9HRE
     1SIDUALS,3X,16HSTAND. RESIDUALS)
C
  200 FORMAT (10I5)
  201 FORMAT (6F13.5)
  202 FORMAT (1H ,10I12)
  203 FORMAT (14X,3(F12.6,7X),F12.6)
  204 FORMAT (6F13.8)
C
```

```
      K=0
    1 READ  200,NP,IY,IRED,MIT,ISC
      IF (NP) 2,3,2
    2 READ  200,M,N
    4 READ  200,(IX(I),I=1,NP)
      IF (IX(1)) 5,1,5
    5 IF (IRED-1) 16,16,17
   17 K=1
      READ 204,SH2,RO
      READ 204,(BETA(I),I=1,NP)
      IRED=IRED-2
   16 IF (IRED)  6,7,6
    7 CALL F2TCH(IY,Y(1))
      JY=IY+1
      CALL F2TCH(JY,Y(11))
      DO 8 I=1,NP
      CALL F2TCH(IX(I),X(1,I))
      JX(I)=IX(I)+1
    8 CALL F2TCH(JX(I),X(11,I))
      GO TO 10
    6 READ 201,(Y(J),J=M,N)
      DO 9  I=1,NP
    9 READ 201,(X(J,I),J=M,N)
   10 PRINT 100,M,N
      PRINT 101
      PRINT 202,IY,(IX(I),I=1,NP)
      PRINT 101
      NT=N-M+1
      DO 11 I=1,NP
      DO 12 J=M,N
      JJ=J-M+1
      A(JJ)=Y(J)
```

```
   12 Z(JJ)=X(J,I)
      DO 13 J=1,NT
      Y(J)=A(J)
   13 X(J,I)=Z(J)
   11 CONTINUE
      CALL MAPRI (4,NP,NT,Y,TEMP,X,70)
C
C
      IF (K) 18,18,19
   18 CALL REGRES (NP,NT,Y,X,SH2,RO,BETA,A,E,V,TE)
      PRINT 103
      DO 14 I=1,NT
   14 PRINT 203,Y(I),A(I),E(I),V(I)
C
C     INITIAL PARAMETER VECTOR
C
C
   19 SDEV=SQRT(SH2)
      TH(1)=SDEV
      TH(2)=RO
      DO 15 I=1,NP
      J=I+2
   15 TH(J)=BETA(I)
C
      CALL LINK (MAIN2)
C
    3 CALL EXIT
      END
```

```
C      MAIN2 PROGRAM - MODSCO-AUTOCORRELATED MODEL
C      SECOND PART
C
       DIMENSION X(70,5),Y(70),E(70),TH(7),TE(5)
C
       COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT,TE,ISC
C
C      CONSTANTS
C
       EPS1=1./10.**5
       EPS2=EPS1
       NK=NP+2
C
       IF (ISC) 21,20,21
   20  CALL MODSCO (NK,EPS1,EPS2)
       GO TO 22
   21  CALL SCOR (NK,EPS2)
   22  CALL LINK (MAIN1)
C
       END
```

```
      SUBROUTINE MODSCO (NK,EPS1,EPS2)
C
      DIMENSION X(70,5),Y(70),E(70),TH(7),TE(5)
      DIMENSION A(70),TB1(7),TB(7),SD(7),SC(7),AA(2,2),B(5,5),BATA(5)
C
      COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT,TE
C
 1000 FORMAT (1H1,44H REV. METH. OF SCORING APPLIED ON AN AUTOCOR,13HREL
     1ATED MODEL,//1H ,I5,14H OBSERVATIONS,I5,34H PARAMETERS (SDEV AND R
     20 INCLUDED))
 1001 FORMAT (/20H INIT. PARAM. VALUES)
 1002 FORMAT (/21H INIT. VALUE CONC. LF,F19.10)
 1003 FORMAT (/////45X,13HITERATION NO.,I4)
 1004 FORMAT (23H TEST POINT PAR. VALUES)
 1005 FORMAT (//30H TEST POINT VALUE OF CONC. LF.,F15.8)
 1006 FORMAT (//27H PAR. VALUES VIA REGRESSION)
 1007 FORMAT (//23H NUMBER OF STEP CHANGES,20X,I15)
 1008 FORMAT (//11H FINAL STEP,32X,F15.8)
 1009 FORMAT (//36H VALUE OF CONC. LF. AFTER REGRESSION,F15.8)
 1010 FORMAT (//39H RELATIVE CHANGE IN EACH PAR. LESS THAN ,E12.4)
 1011 FORMAT (//48H RELATIVE CHANGE IN VALUE OF CONC. LF. LESS THAN,E12.
     14)
 1020 FORMAT (1H1,25H FINAL ADDITIONAL INFORM.)
 1021 FORMAT (11H  Y-VALUES ,7X,8HOBSERVED,9X,10HCALCULATED,10X,9HRESIDU
     1ALS)
 1022 FORMAT (/,21H FINAL INFORM. MATRIX)
 1023 FORMAT (////)
 1024 FORMAT (/,33H COVARIANCE MATRIX OF SDEV AND RO)
 1025 FORMAT (/,33H COVARIANCE MATRIX OF BETA-COEFF.)
 1026 FORMAT (//,4H END)
```

```
 1027 FORMAT (//,11H PAR. ERROR)
 1028 FORMAT (//,19H IDEC OR IINC ERROR)
 1029 FORMAT (//,48H THE VALUE OF THE CONC. LF. CANNOT BE REDUCED TO,37H
     2THAT AT THE END OF THE LAST ITERATION)
 1030 FORMAT (//,30H A (RELATIVE) MAX. IS OBTAINED)
 1031 FORMAT (//,9H T-VALUES)
C
 2000 FORMAT (14X,2(F12.6,7X),F12.6)
C
      EPS3=1./10.**5
      T=NT
      PRINT 1000,NT,NK
      PRINT 1001
      CALL  MAPRI (1,NK,ITEMP,TH,TEMP,TEMP,ITEMP)
      IF (NT-NK) 99,99,15
   15 IF (NK-7) 16,16,99
   16 IF (NK-1)  99,17,17
   17 IF (MIT-1) 99,18,18
   18 IF (MIT-1000) 19,19,99
   19 DO 20 I=1,NK
      IF (TH(I)) 20,99,20
   20 CONTINUE
      IRAN=1
      JORDAN=1
      IF(EPS1) 5,10,10
    5 EPS1=0
   10 IF(EPS2) 40,40,30
   40 IF(EPS1) 60,60,50
   60 IRAN=2
      GO TO 70
   50 IRAN=3
      GO TO 70
```

```
   30 IF(EPS1) 80,80,70
   80 JORDAN=2
   70 CALL MODT (NK,TH,VAL1)
      PRINT 1002,VAL1
      NIT=1
C                                                  BEGIN ITERATION
  100 PRINT 1003,NIT
C
C     INFORMATION MATRIX
      CALL INFO (NK,AA,B)
C
C     SCORE VECTOR
C
      SH2=TH(1)**2
      RO2=TH(2)**2
      SC(1)=-T/TH(1)+FGH/TH(1)**3
      SC(2)=-TH(2)/(1.-RO2)-(TH(1)*G-H)/SH2
      DO 111 I=1,NP
      Q =0.
      Q3=0.
      Q2=0.
      Q1=0.
      QT=0.
      FF=0.
      HH1=0.
      HH=0.
      B1=0.
      BT=0.
      DO 109 L=1,NT
  109 CALL FLEM (FF,Q,E(L),X(L,I))
```

```
      DO 110 L=2,NT
      JJ=L-1
      CALL FLEM (HH1,Q3,X(L,I),E(JJ))
  110 CALL FLEM (HH,Q2,E(L),X(JJ,I))
      CALL FLEM (B1,Q1,E(1),X(1,I))
      CALL FLEM (BT,QT,E(NT),X(NT,I))
      J=I+2
      SC(J)=(FF+RO2*(FF-B1-BT)-TH(2)*(HH+HH1))/SH2
  111 CONTINUE
C
C
C     STEEPEST ASCENT
      DO 112 I=1,2
      DO 112 J=1,2
      Q=0.
      SD(I)=0.
  112 CALL FLEM (SD(I),Q,AA(I,J),SC(J))
      DO 113 I=1,NP
      Q=0.
      JJ=I+2
      SD(JJ)=0.
      DO 113 L=1,NP
      J=L+2
      CALL FLEM (SD(JJ),Q,B(I,L),SC(J))
  113 CONTINUE
C
C
C     COMPUTE OPTIMAL STEP
C
      IC=1
      STEP=1.
```

```
      STEP1=STEP
  169 IDEC=0
      IINC=0
      GO TO 171
  170 SAL=VAL
      DO 850 I=1,NK
  850 TB1(I)=TB(I)
  171 DO 220 I=1,NK
  220 TB(I)=TH(I)+STEP*SD(I)
      PRINT 1004
      CALL MAPRI (1,NK,ITEMP,TB,TEMP,TEMP,ITEMP)
C
      CALL MODT (NK,TB,VAL)
      PRINT 1005,VAL
C
      IF (IINC-1) 153,152,720
  153 IF (IDEC) 720,154,154
  152 IF ((VAL-SAL)/ABS(SAL)-EPS1) 661,661,663
C
  154 IF ((VAL-VAL1)/ABS(VAL1)-EPS1) 661,661,663
  663 IINC=1
      IF (IDEC) 720,721,662
  721 STEP1=STEP
      STEP=STEP*2.
  750 IC1=IC
      IC=IC+1
      IF (IC-36) 170,2700,2700
  661 IDEC=1
      IF (IINC) 720,722,664
  722 STEP=STEP/2.
      STEP1=STEP
      IF (STEP-EPS3) 150,150,750
```

```
  150 STEP=0.
      GO TO 280
  664 DO  668 I=1,NK
  668 TB(I)=TB1(I)
      IC=IC1
      VAL=SAL
      STEP=STEP1
  662 PRINT 1006
      CALL MAPRI (1,NK,ITEMP,TB,TEMP,TEMP,ITEMP)
      PRINT 1007,IC
      PRINT 1008,STEP
      PRINT 1009,VAL
      GO TO  (225,270,265),IRAN
  225 DO 240 I=1,NK
      IF (ABS(STEP*SD(I)/TH(I))-EPS2) 240,240,250
  240 CONTINUE
      PRINT 1010,EPS2
      GO TO 280
  250 GO TO (265,270),JORDAN
  265 IF (ABS((VAL-VAL1)/VAL1)-EPS1)  260,260,270
  260 PRINT 1011,EPS2
      GO TO 280
  270 VAL1=VAL
      DO 669 I=1,NK
  669 TH(I)=TB(I)
C
      NIT=NIT+1
      IF (NIT-MIT) 100,100,280
  280 PRINT 1030
      PRINT 1006
```

```
      CALL MAPRI (1,NK,ITEMP,TH,TEMP,TEMP,ITEMP)
      DO 180 J=1,NP
      JJ=J+2
  180 BATA (J)=TH(JJ)/(TH(1)*SQRT(TE(J)))
      PRINT 1031
      CALL MAPRI (1,NP,ITEMP,BATA,TEMP,TEMP,ITEMP)
      PRINT 1009,VAL1
      PRINT 1020
      PRINT 1021
      CALL MRES (NK,TH,A)
      DO 143  I=1,NT
  143 PRINT 2000,Y(I),A(I),E(I)
C
C     COVARIANCE-MATRIX VIA INFORMATION MATRIX
C
      CALL INFO (NK,AA,B)
      PRINT 1022
      PRINT 1023
      PRINT 1024
      CALL MAPRI (3,2,ITEMP,TEMP,TEMP,AA,2)
      PRINT 1025
      CALL MAPRI (3,NP,ITEMP,TEMP,TEMP,B,5)
  410 PRINT 1026
      RETURN
   99 PRINT 1027
      GO TO 410
  720 PRINT 1028
      GO TO 410
 2700 PRINT 1029
      PRINT 1007,IC
      PRINT 1008,STEP
      GO TO 410
      END
```

```
      SUBROUTINE MODT (NK,PAR,VAL)
C
      DIMENSION X(70,5),Y(70),E(70),TH(7)
      DIMENSION A(70),PAR(7)
C
      COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT
C
      CALL MRES (NK,PAR,A)
      CALL HULP (PAR)
      T=NT
      VAL=(LOG((1.-PAR(2)**2)/PAR(1)**(2.*T)))/2.-FGH/(2.*(PAR(1)**2))
      RETURN
      END
```

```
      SUBROUTINE MRES (NK,PAR,A)
C
      DIMENSION X(70,5),Y(70),E(70),TH(7)
      DIMENSION A(70),PAR(7)
C
      COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT
C
      DO 1 J=1,NT
      A(J)=0.
      Q=0.
      DO 2 I=1,NP
      K=I+2
    2 CALL FLEM (A(J),Q,X(J,I),PAR(K))
    1 E(J)=Y(J)-A(J)
      RETURN
      END
```

```
      SUBROUTINE HULP (PAR)
C
      DIMENSION X(70,5),Y(70),E(70),TH(7)
      DIMENSION PAR(7)
C
      COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT
C
      H=0.
      F=0.
      QF=0.
      QH=0.
      DO 1  J=1,NT
    1 CALL FLEM (F,QF,E(J),E(J))
      DO 2  J=2,NT
      K=J-1
    2 CALL FLEM (H,QH,E(J),E(K))
      G1=0.
      GT=0.
      Q1=0.
      QT=0.
      CALL FLEM (G1,Q1,E(1),E(1))
      CALL FLEM (GT,QT,E(NT),E(NT))
      G=F-G1-GT
      FGH=F+(PAR(2)**2)*G-2.*PAR(2)*H
      RETURN
      END
```

```
      SUBROUTINE INFO (NK,A,B)
C
      DIMENSION X(70,5),Y(70),E(70),TH(7)
      DIMENSION A(2,2),B(5,5)
C
      COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT
C
  100 FORMAT (/23H NO INVERSE OF B-MATRIX)
  101 FORMAT (/17H NP IN INFO WRONG)
C
C
C     INFORMATION MATRIX  I= (A 0)
C                            (0 B)
C
C     WHERE A IS A 2*2 MATRIX
C
C     INVERSE OF A
C
      SH2=TH(1)**2
      RO2=TH(2)**2
      T=NT
      DEN=(T+RO2*(T-2.))/(1.-RO2)+T**2-2.*T
      A(1,1)=SH2*((1.+RO2)/(1.-RO2)+T-2.)/(2.*DEN)
      A(1,2)=TH(1)*(-TH(2))/DEN
      A(2,1)=A(1,2)
      A(2,2)=T*(1.-RO2)/DEN
C
C     CONSTRUCTION OF B
C
      DO 1 I=1,NP
      DO 1 J=1,I
      QC=0.
      Q3=0.
      QD=0.
```

```
      Q1=0.
      QT=0.
      C=0.
      DD=0.
      D=0.
      B1=0.
      BT=0.
      DO 2 L=1,NT
    2 CALL FLEM (C,QC,X(L,I),X(L,J))
      DO 3 L=2,NT
      K=L-1
      CALL FLEM (DD,Q3,X(K,I),X(L,J))
    3 CALL FLEM (D,QD,X(L,I),X(K,J))
      CALL FLEM (B1,Q1,X(1,I),X(1,J))
      CALL FLEM (BT,QT,X(NT,I),X(NT,J))
      B(I,J)=(C+RO2*(C-B1-BT)-TH(2)*(D+DD))/SH2
    1 B(J,I)=B(I,J)
C
C     INVERSE B-MATRIX
C
      IF (NP-1) 4,7,8
    4 PRINT 101
      GO TO 9
    7 B(1,1)=1./B(1,1)
      GO TO 6
    8 DET=0.0000001
      CALL INVE05 (B,NP,DET)
      IF (NP) 5,5,6
    5 PRINT 100
      NP=-NP
    9 STOP
    6 RETURN
      END
```

```
      SUBROUTINE SCOR (NK,EPS1)
C
      DIMENSION X(70,5),Y(70),E(70),TH(7),TE(5)
      DIMENSION A(70),TB1(7),TB(7),SD(7),SC(7),AA(2,2),B(5,5),BATA(5)
C
      COMMON X,Y,E,G,H,FGH,TH,NP,NT,MIT,TE
C
 1000 FORMAT (1H1,39H METH. OF SCORING APPLIED ON AN AUTOCOR,13HRELATED
     1MODEL,//1H ,I5,14H OBSERVATIONS,I5,34H PARAMETERS (SDEV AND RO INC
     2LUDED))
 1001 FORMAT (/20H INIT. PARAM. VALUES)
 1003 FORMAT (/////45X,13HITERATION NO.,I4)
 1004 FORMAT (23H TEST POINT PAR. VALUES)
 1005 FORMAT (//30H TEST POINT VALUE OF CONC. LF.,F15.8)
 1006 FORMAT (//27H PAR. VALUES VIA REGRESSION)
 1009 FORMAT (//36H VALUE OF CONC. LF. AFTER REGRESSION,F15.8)
 1010 FORMAT (//39H RELATIVE CHANGE IN EACH PAR. LESS THAN ,E12.4)
 1020 FORMAT (1H1,25H FINAL ADDITIONAL INFORM.)
 1021 FORMAT (11H  Y-VALUES ,7X,8HOBSERVED,9X,10HCALCULATED,10X,9HRESIDU
     1ALS)
 1022 FORMAT (/,21H FINAL INFORM. MATRIX)
 1023 FORMAT (////)
 1024 FORMAT (/,27H COV. MATRIX OF SDEV AND RO)
 1025 FORMAT (/,27H COV. MATRIX OF BETA-COEFF.)
 1026 FORMAT (//,4H END)
 1027 FORMAT (//,11H PAR. ERROR)
 1030 FORMAT (//,30H A (RELATIVE) MAX. IS OBTAINED)
 1031 FORMAT (//,9H T-VALUES)
 1032 FORMAT (//,19H VALUE OF CONC. LF.,F15.8)
 1033 FORMAT (//,30H TEST POINT VALUE OF CONC. LF.,F15.8)
```

```
 2000 FORMAT (14X,2(F12.6,7X),F12.6)
C
      T=NT
      PRINT 1000,NT,NK
      PRINT 1001
      CALL  MAPRI (1,NK,ITEMP,TH,TEMP,TEMP,ITEMP)
      IF (NT-NK) 99,99,15
   15 IF (NK-7) 16,16,99
   16 IF (NK-1)  99,17,17
   17 IF (MIT-1) 99,18,18
   18 IF (MIT-1000) 19,19,99
   19 DO 20 I=1,NK
      IF (TH(I)) 20,99,20
   20 CONTINUE
      CALL MODT (NK,TH,VAL)
      PRINT 1032,VAL
C                                                   BEGIN ITERATION
      NIT=1
  100 PRINT 1003,NIT
C
C     INFORMATION MATRIX
      CALL INFO (NK,AA,B)
C
C     SCORE VECTOR
C
      CALL MRES (NK,TH,A)
      CALL HULP (TH)
      SH2=TH(1)**2
      RO2=TH(2)**2
      SC(1)=-T/TH(1)+FGH/TH(1)**3
```

```
      SC(2)=-TH(2)/(1.-RO2)-(TH(1)*G-H)/SH2
      DO 111 I=1,NP
      Q =0.
      Q3=0.
      Q2=0.
      Q1=0.
      QT=0.
      FF=0.
      HH1=0.
      HH=0.
      B1=0.
      BT=0.
      DO 109 L=1,NT
  109 CALL FLEM (FF,Q,E(L),X(L,I))
      DO 110 L=2,NT
      JJ=L-1
      CALL FLEM (HH1,Q3,X(L,I),E(JJ))
  110 CALL FLEM (HH,Q2,E(L),X(JJ,I))
      CALL FLEM (B1,Q1,E(1),X(1,I))
      CALL FLEM (BT,QT,E(NT),X(NT,I))
      J=I+2
      SC(J)=(FF+RO2*(FF-B1-BT)-TH(2)*(HH+HH1))/SH2
  111 CONTINUE
C
C     STEEPEST ASCENT
C
      DO 112 I=1,2
      DO 112 J=1,2
      Q=0.
      SD(I)=0.
```

```
  112 CALL FLEM (SD(I),Q,AA(I,J),SC(J))
      DO 113 I=1,NP
      Q=0.
      JJ=I+2
      SD(JJ)=0.
      DO 113 L=1,NP
      J=L+2
      CALL FLEM (SD(JJ),Q,B(I,L),SC(J))
  113 CONTINUE
C
C
C     COMPUTE OPTIMAL STEP
C
      STEP=1.
      DO 220 I=1,NK
  220 TB(I)=TH(I)+STEP*SD(I)
      PRINT 1004
      CALL MAPRI (1,NK,ITEMP,TB,TEMP,TEMP,ITEMP)
      CALL MODT (NK,TB,VAL)
      PRINT 1033,VAL
C
      DO 240 I=1,NK
      IF (ABS(STEP*SD(I)/TH(I))-EPS1) 240,240,270
  240 CONTINUE
      PRINT 1010,EPS2
      GO TO 280
  270 DO 669 I=1,NK
  669 TH(I)=TB(I)
C
      NIT=NIT+1
      IF (NIT-MIT) 100,100,280
  280 PRINT 1030
```

```
      PRINT 1006
      CALL MAPRI (1,NK,ITEMP,TH,TEMP,TEMP,ITEMP)
      DO 180 J=1,NP
      JJ=J+2
  180 BATA (J)=TH(JJ)/(TH(1)*SQRT(TE(J)))
      PRINT 1031
      CALL MAPRI (1,NP,ITEMP,BATA,TEMP,TEMP,ITEMP)
      CALL MODT (NK,TH,VAL)
      PRINT 1009,VAL
      PRINT 1020
      PRINT 1021
      CALL MRES (NK,TH,A)
      DO 143  I=1,NT
  143 PRINT 2000,Y(I),A(I),E(I)
C
C     COVARIANCE-MATRIX VIA INFORMATION MATRIX
C
      CALL INFO (NK,AA,B)
      PRINT 1022
      PRINT 1023
      PRINT 1024
      CALL MAPRI (3,2,ITEMP,TEMP,TEMP,AA,2)
      PRINT 1025
      CALL MAPRI (3,NP,ITEMP,TEMP,TEMP,B,5)
  410 PRINT 1026
      RETURN
   99 PRINT 1027
      GO TO 410
      END
```

E.I.T. 1970